

Using IBM TotalStorage Enterprise Storage  
Server (ESS)<sup>®</sup> with  
IBM<sup>®</sup> DB2<sup>®</sup> UDB Enterprise Edition V7.2  
and  
SAP<sup>®</sup> R/3<sup>®</sup>

October 2002

Brigitte Bläser – IBM Germany Böblingen Laboratory  
Enzo Cialini – IBM Toronto Laboratory  
Aslam Nomani – IBM Toronto Laboratory  
David Davis – IBM Storage Systems Group



---

## Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX  
DB2  
DB2 Universal Database  
Enterprise Storage Server (ESS)  
ESCON  
FlashCopy  
IBM  
iSeries  
zSeries

The following terms are trademarks SAP AG in Germany, in the United States, other countries, or both:

SAP  
mySAP.com  
R/3

Other products or service names may be trademarks or registered trademarks of their respective owners.

Unix and Unix-based trademarks and logos are trademarks or registered trademarks of The Open Group.  
Windows and Windows-based trademarks and logos are trademarks or registered trademarks of Microsoft Corp.

The furnishing of this document does not imply giving license to any IBM patents.

References in this document to IBM products, Programs, or Services do not imply that IBM intends to make these available in all countries in which IBM operates.

The information contained in this publication does not include any product warranties, and any statements provided in this document should not be interpreted as such.

---

## Table of contents

ABSTRACT.....	IV
ABOUT THE AUTHORS .....	V
DB2 UDB SUSPENDED I/O OVERVIEW.....	1
DB2 UDB COMMANDS <<FOR CONSISTENCY, I COPIED THESE FROM THE EMC PAPER>> .....	2
SET WRITE SUSPEND .....	2
SET WRITE RESUME .....	2
DB2INIDB.....	2
OVERVIEW OF IBM 2105 ENTERPRISE STORAGE SERVER .....	4
OVERVIEW OF COPY SERVICES .....	5
OVERVIEW OF THE ESS COPY SERVICES WEB INTERFACE .....	7
OVERVIEW OF ESS COMMAND LINE INTERFACE (CLI) .....	8
ESS COPY SERVICES CLI COMMAND DESCRIPTION .....	9
RSEXECUTETASK.....	9
RSLIST2105S .....	9
RSPRIMESERVER .....	9
RSQUERY .....	9
RSTESTCONNECTION.....	9
RSQUERYCOMPLETE .....	9
OVERVIEW OF STANDARD SAP R/3 SYSTEM INSTALLATION WITH DB2 UDB.....	10
SAMPLE CONFIGURATION OF SAP R/3 WITH DB2 UDB AND IBM ESS .....	12
CONFIGURING A SYSTEM WITH TWO HOSTS.....	15
<i>Overview</i> .....	15
SNAPSHOT DATABASE SCENARIO .....	26
STANDBY DATABASE SCENARIO.....	28
SPLIT-MIRROR ONLINE BACKUP SCENARIO .....	31
SPLIT-MIRROR RESTORE SCENARIO.....	3433
SUMMARY .....	3534

---

## Abstract

SAP® R/3™ and its components have been the success of more than 19,000 installations in enterprises of every size in more than 106 countries worldwide. The SAP R/3 system is an open, integrated system that covers all business applications including accounting and controlling, production and materials management, quality management and plant maintenance, sales and distribution, human resources and project management. The mySAP.com™ e-business platform is a suite of solutions and services that empower employees, customers, and business partners to collaborate successfully - anywhere, anytime.

IBM DB2 Universal Database™ (UDB) is the industry's first multimedia, Web-ready relational database management system, strong enough to meet the demands of large corporations and flexible enough to serve medium-sized and small e-businesses. DB2 Universal Database combines integrated power for business intelligence, content management, and e-business with industry-leading performance and reliability and is thus ideally suited as the database system for SAP R/3 and mySAP.com.

This coupled with the FlashCopy® feature of the IBM Enterprise Storage Server® enables high availability solutions, like the creation of clone SAP R/3 systems, standby databases and system backups with nearly no performance impact and zero downtime.

This document takes you through a sample configuration setup using SAP R/3 Release 4.6D with DB2 UDB Enterprise Edition (EE) V7.2 and IBM Enterprise Storage Server FlashCopy.

---

## About the authors

**Aslam Nomani** has been working with the IBM Database Technology team in the Toronto Laboratory for five years. For the past four years, he has worked in the DB2 UDB System Verification Test department. Aslam has worked extensively in testing DB2 UDB EE and DB2 UDB EEE in a high availability environment. He is currently one of the team leads within the UDB test organization.

**Enzo Cialini** has been working with Database Technology at the IBM SWS Toronto Laboratory for ten years, is certified in DB2 UDB Database Administration and DB2 UDB Database Application Development, is an Advanced Technical Expert in DB2 for DRDA, and an Advanced Technical Expert in DB2 for Clusters. He is currently responsible for managing the DB2 UDB System Verification Test Department, with a focus on High Availability, and has been involved with DB2 in highly available environments for many years. His experience ranges from implementing and supporting numerous installations to consulting.

**Brigitte Bläser** has been working for IBM for 17 years. She has been engaged in several development projects using DB2 UDB and SAP R/3. She is certified in DB2 UDB Database Administration and DB2 UDB Database Application Development. She is currently working in the SAP DB2 Development/Porting department of the IBM Germany Böblingen Lab and responsible for supporting the SAP IT department in migrating internal SAP systems to DB2 UDB. This includes the definition and implementation of High Availability solutions.

**David Davis** is a Product Professional with SSG Software Development in San Jose, California. He has 1 year of experience in application and database integration with storage products. Previously he worked for 6 years in IBM's Professional Services as a consultant in database and application development engagements.

---

## DB2 UDB suspended I/O overview

DB2 UDB v7.2 has implemented suspended I/O technology, which provides the capability to use split mirroring technology while DB2 is online. Suspended I/O supports continuous system availability by providing a full implementation for splitting a mirror (pair) without shutting down the database. The split copy of the database can be used to do such things as:

- Provide a transactionally consistent snapshot of the database at the current point in time. This database can be used to offload user queries that don't need the most current version of the database.
- Provide a standby database that can be accessed as a disaster recovery strategy if the primary database is not available. All logs from the primary database will be applied to the secondary database so that the secondary database represents a transactionally consistent version of the primary database at the time of the failure.
- Provide the ability to offload backups from the primary database. A DB2 backup can be performed on the secondary database system. The DB2 backup can then be restored on either the primary system or on another system. Then the database can be rolled forward to bring the database to a particular point in time, or until the end of the logs are reached.
- Provide the ability for a quick database restore. The mirror (pair) can be resynced so that the primary copy is restored to the initial data at the time of the split. Then the primary database can be rolled forward to bring the database to a particular point in time, or until the end of the logs are reached.

---

## DB2 UDB commands

We use the following commands:

### **set write suspend**

The suspend command (db2 set write suspend for database) suspends all write operations to the DB2 UDB database (i.e., to tablespaces and log files). Read operations are not suspended and are thus allowed to continue. Applications can continue to process insert, update, and delete operations utilizing the DB2 bufferpools. A database connection is required for issuing the suspend command. It is recommended that you maintain the current session for executing the subsequent resume command.

### **set write resume**

The resume command (db2 set write resume for database) resumes all write operations to the suspended DB2 UDB database. A database connection is required for issuing the resume command.

### **db2inidb**

The db2inidb command (db2inidb <db\_alias> as < snapshot | standby | mirror >) is required to initialize the copy of the suspended database. You do not require a database connection prior to executing this command. It can be used in the following three cases:

- **snapshot** can be applied to the secondary copy, putting it into a transactionally consistent state.
- **standby** can be applied to the secondary copy, putting it into a rollforward pending state. DB2 logs from the primary database can then be applied to the secondary database.
- **mirror** can be applied to the primary copy after it has been restored from the secondary copy. The primary database will be placed into a rollforward pending state and then DB2 logs can be applied to the primary database.

### **restart write resume**

The restart write resume command (db2 restart db <db\_alias> write resume) can be utilized to resume write operations on a suspended database that experienced an abnormal termination.

For further information regarding DB2 UDB's suspended I/O functionality, see the *DB2 UDB documentation* and *Release Notes* provided with V7.2 or V7.1 Fixpack 3 at [http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7pubs.d2w/en\\_main](http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7pubs.d2w/en_main). In addition, see the DB2 Developer Domain article entitled *Split Mirror Using Suspended I/O in DB2*



*Universal Database V7*

<http://www7b.software.ibm.com/dmdd/library/techarticle/0204quazi/0204quazi.html..>

---

## Overview of IBM 2105 Enterprise Storage Server

The IBM 2105 Enterprise Storage Server (ESS) provides superior storage sharing for Unix®, Windows NT®, Windows 2000®, Novell Netware, iSeries™ and zSeries™ servers. With the ESS any combination of these heterogeneous platforms can be used. Hosts are attached to the ESS via the following interfaces:

- Fibre Channel (for Unix, Windows NT and Windows 2000),
- UltraSCSI (for Unix, Windows NT and Windows 2000, Novell Netware, and iSeries)
- ESCON and FICON (for zSeries)

The ESS provides the image of a set of logical disk devices to the attached servers. The logical devices are configured to emulate disk device types that are compatible with the attached servers. The logical devices access a logical volume that is implemented using multiple disk drives.

The ESS consists of the following components:

- Storage server: the storage server is composed of two clusters that control and manage data transfer. If one cluster fails the remaining cluster takes over the functions of the failing one.
- Host adapters: each cluster has one or more host adapters that provide one or more host I/O interfaces. A host adapter can communicate with either cluster complex
- Device adapters: each cluster has one or more device adapters that provide one or more storage device interfaces. Disk drives are attached to a pair of device adapters, one in each cluster, so that the drives are accessible from either cluster.
- Cluster complex: the cluster complex provides the management functions for the ESS. It consists of cluster processors, cluster memory, cache, nonvolatile storage and related logic
- The disk drives provide the primary nonvolatile storage medium for any host data stored within the ESS storage devices. They are grouped into ranks and are managed by the clusters.

Internally the ESS is divided into Logical Subsystems (LSSs). An LSS is a logical structure that is used for configuration of the ESS. An ESS can have up to 16 LSSs defined for Open Systems storage. In one LSS, up to 256 logical volumes can be defined.

Storage capacity is partitioned among the attached servers using the flexible, Web-based StorWatch Enterprise Storage Specialist management tool. With its Web interface you can create, assign, and unassign logical volumes to the hosts attached to the ESS.

For details about the ESS, ESS Copy Services, and the Web and command line interfaces please refer to the following Redbooks available at <http://www.redbooks.ibm.com>:

- SG245757 *Implementing ESS Copy Services on UNIX and Windows NT/2000*
- SG245465 *IBM Enterprise Storage Server*

---

## Overview of Copy Services

Copy Services is a separately priced feature of the ESS, which provides powerful functions for replication of mission-critical data. For the Open Systems environment Copy Services offer point-in-time replication with FlashCopy and dynamic synchronous mirroring to a remote ESS with Peer-to-Peer Remote Copy (PPRC). The examples described in this paper use FlashCopy.

FlashCopy creates instant point-in-time copies of logical volumes (LVs) for purposes such as data backup or data duplication. The copy is called time-zero (T0) copy. It is a binary copy that looks exactly the same as the source volume at the time the copy was made.

When executing FlashCopy the ESS establishes the FlashCopy pairs and creates control bitmaps. This only takes a few seconds. Both source and target volumes can be accessed immediately after the FlashCopy pairs have been established, and the bitmaps created. Background tasks continue copying the data from the source to the target volumes. The FlashCopy relationship between source and target volume ends when the physical background copy tasks completed. Source and target volume can be accessed and modified independently.

FlashCopy can also be used with the 'no copy' option. This suppresses copying of the complete source volume. Only the data that changes on the source volume is copied to the target volume before the changes are made on the source volume. The FlashCopy relationship between source and target volume exists until it is withdrawn manually. When the FlashCopy relationship ends, the copy is no longer available for use.

Here are the restrictions for FlashCopy:

- A FlashCopy relationship can only be established between disk volumes. Source and target volumes have to reside within the same Open System's Logical Subsystem (LSS).
- The target disk must have the same size as the source volume or must be bigger.
- Both source and target volumes can only be involved in one FlashCopy relationship at a time
- The target can only be used on the same operating system.

Copy Services run inside the ESS. FlashCopy can be used to quickly create copies of production data for the purpose of taking backups, creating test systems, or creating database copies for data mining purposes or data warehouses. FlashCopy has no significant impact on the performance nor does it require downtime of the production system.

Using PPRC, one ESS has to be defined as the Primary Copy Services Server. This ESS holds all Copy Services related information (i.e. the volumes and their current state). Optionally a second ESS could be defined to be the Backup Copy Services Server. On each ESS that is intended to use Copy Services there is a Copy Services client running which communicates with the server. Access to Copy Services is provided through an Internet browser. This allows for controlling the ESS copy functionality over the network from any platform for which the browser is supported.



---

## Overview of the ESS Copy Services Web interface

The ESS Copy Services Web interface is used to create and manage Copy Services tasks. This Web interface consists of a number of HTML screens and Java™ applets and can be accessed from a standard Web browser.

A Copy Services task consists of:

- The IDs of the source and target disks involved,
- The operation to be performed:
  - Establish FlashCopy pair
  - Withdraw FlashCopy pair
  - Establish PPRC copy pair
  - Suspend PPRC copy pair
  - Terminate PPRC copy pair
- Several other options (that is, . whether or not to perform background copy).

For example, to create a task named `fctask` that establishes a FlashCopy relationship between two disks with the 'no copy' option:

1. Select the LSS in which the source and target disk reside.
2. Left click the source disk, right click the target disk, and right click again to display the **Options** menu.
3. Select **Establish FlashCopy pair** from the first menu and click **Next**.
4. Select **Do not perform background copy if checked** from the second menu and click **Next**.
5. Enter `fctask` as the task's name and a task description and click **Save** to store the task on the Copy Services server.

You can execute the task from the **Tasks** menu of the Copy Services Web interface, or from any server on which the ESS Command Line interface is installed. You can install the command line interface on any server attached to the ESS via TCP/IP.

A basic FlashCopy task can only contain disk pairs from one LSS. You can combine several basic FlashCopy tasks into one group task to establish or withdraw FlashCopy pairs in several LSSs. Please note that the source and target disks of a FlashCopy pair always have to reside in the same LSS.

---

## Overview of ESS Command Line Interface (CLI)

The ESS Command Line Interface consists of a number of scripts to communicate with the ESS Copy Services server. The Command Line Interface is Java-based and therefore requires the Java JDK version 1.1.8 to be installed on each host system from which you want to issue the commands. The CLI code level must be at the same release level as the microcode that is installed in the ESS clusters. The CLI can be installed on any host system that is connected to the ESS that is defined as the primary Copy Services server via a local area network. It is not required that the host system is attached to the ESS storage.

In the ESS Copy Services Web Interface you can configure that the execution of Copy Services server commands is password protected. If you enable password protection you have to provide a user ID and password with each command invocation.

---

## ESS Copy Services CLI command description

### **rsExecuteTask**

Executes one or more predefined Copy Services Server tasks and waits for the tasks to complete execution. The command returns 0 for successful completion and greater-than zero if there is an error. .

**Note:** For FlashCopy tasks, execution is completed when the FlashCopy pairs have been established and the control bitmaps have been generated. This usually only lasts a few seconds.

### **rsList2105s**

Displays the mapping of host physical volume name to ESS volume serial number for all disks assigned to the host system from which the command is executed.

### **rsPrimeServer**

Notifies the Copy Services Server of the mapping of host disk name to ESS volume serial number. This permits a host volume view from the ESS Copy Services Server Web interface.

### **rsQuery**

Queries the FlashCopy and PPRC status of one or more volumes.

### **rsTestConnection**

Determines whether the Copy Services Server can be contacted successfully.

### **rsQueryComplete**

Determines PPRC status.

With the exception of rsList2105s, all commands require the IP address or complete hostname of the Primary Copy Services Server as parameter. Optionally, you can specify the address of the Backup Copy Services Server.

---

## Overview of standard SAP R/3 system installation with DB2 UDB

A SAP R/3 system is identified by a three character system identifier <SID> and a two digit system number between 00 and 96.

The following information about the directory structure applies to a Central System consisting of the SAP R/3 central instance and database instance installed on a single host. We do not consider the installation of dialog instances on additional systems. We will further assume that the transport host is installed on the same host as the central instance.

The base directories required for the SAP R/3 central instance are /usr/sap for the general SAP R/3 software, and /usr/sap/trans, which contains SAP R/3 software for the transport of objects between SAP R/3 systems. The SAP R/3 software and configuration files are installed in the directory /usr/sap/<SID>. The directory is owned by the user <sid>adm, which belongs to the group sapsys. For each R/3 system there is one DB2 instance db2<sid> that contains the SAP R/3 database <SID> and an administration database ADM<SID> for managing backup images and log files. The db2<sid> directory is owned by the user db2<sid>, which belongs to the group db<sid>adm. db<sid>adm is the SYSADM\_GROUP of the DB2 instance.

This results in the following directory structure:

### SAP software

```
/usr/sap
      <SID>
      trans
```

### DB2 instance

```
/db2/<SID>
  db2<sid>/NODE0000
  log_dir
  log_archive
  log_retrieve
  sapdata1
  sapdata2
  sapdata3
  sapdata4
  sapdata5
  sapdata6
  sapdatat
  sqllib
```

By default all the tablespaces are database-managed storage (DMS), except for the temporary tablespace PSAPTEMP which is system-managed storage (SMS). If you want to use the split mirror copy of the database to take a backup, you should create PSAPTEMP as a DMS tablespace. With DB2 UDB V7.2, you cannot take backups of split mirror database copies unless all the tablespaces are DMS.



Detailed discussions about the layout of a SAP R/3 system on an ESS are contained in the following white papers:

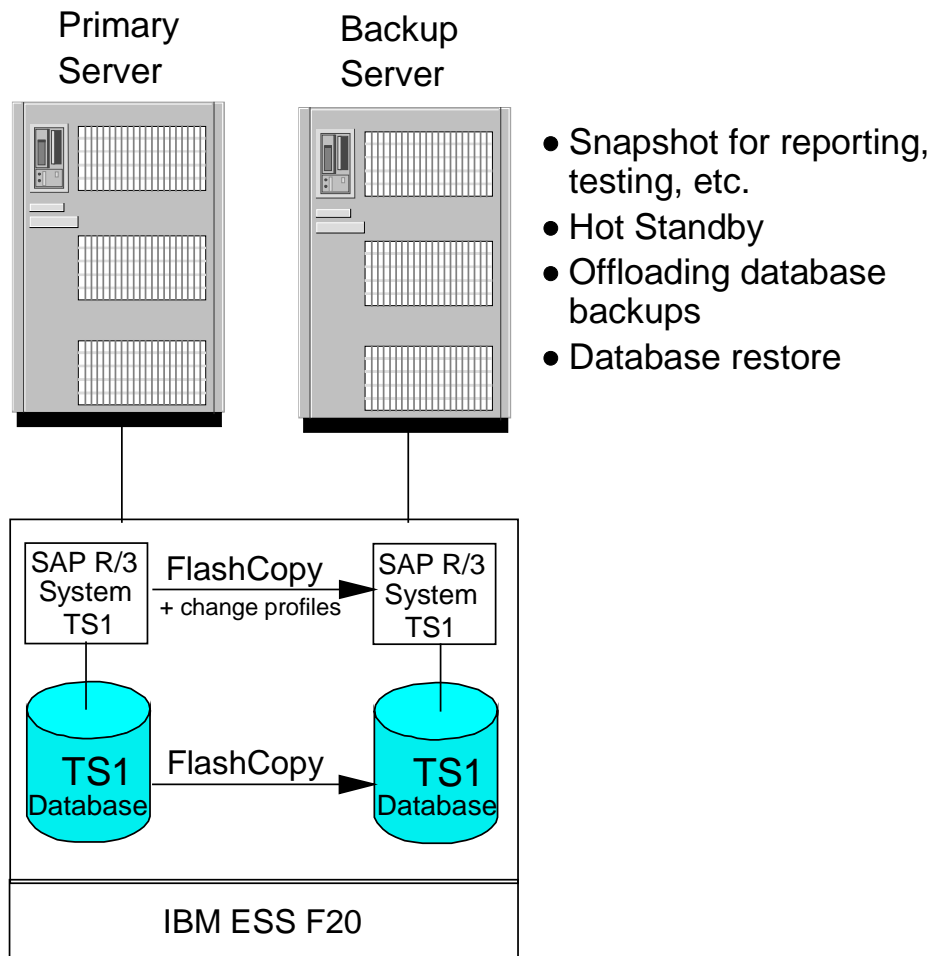
- Sanjoy Das, Siegfried Schmidt, Jens Claussen, BalaSanni Godavari: “Storage Management for SAP and DB2 UDB: Split Mirror Backup / Recovery with IBM Enterprise Storage Server (ESS)”, at <http://www.storage.ibm.com/hardsoft/products/sap/smdb2.pdf>
- Siegfried Schmidt: “SAP R/3 Storage Management – Split Mirror Backup Recovery on IBM’s Enterprise Storage Server (ESS)”, at <http://www.storage.ibm.com/hardsoft/products/sap/smsm200.pdf>
- Jens Claussen: “Database Layout for SAP Installations with DB2 UDB for Unix and Windows”, at <http://service.sap.com/atq>

## Sample configuration of SAP R/3 with DB2 UDB and IBM ESS

In this configuration, we will work with a standard SAP R/3 installation and describe the following different scenarios that use the DB2 UDB split mirror functionality:

- Creating a clone SAP system (**Snapshot database scenario** on page 26)
- Creating a standby SAP database (**Standby database scenario** on page 28)
- Creating a split mirror copy of the SAP R/3 database for backup and restore (**Split-Mirror online backup scenario** on page 31)
- Creating a split mirror copy of the SAP R/3 database that can be copied over the primary database (**Split-mirror restore scenario** on page [3433](#) )

Figure 1 provides the conceptual overview of the sample configuration.



**Figure 1. Conceptual overview**

As Figure 1 shows, there are two AIX hosts: one as the Primary Server, the other as the Backup server.

Ten ESS logical volumes are attached to each system. One logical volume is used for the SAP system software and configuration files, the others for the database. Two

volume groups are created on the Primary server for the SAP system and the database, respectively.

DB2 Universal Database v7.2 for AIX® is used; DB2 UDB suspended I/O capabilities are available on all platforms supported by DB2 UDB.

An IBM ESS 2105 F20 is used. The same scenarios also pertain to other ESS models supporting the same functionality as the ESS F20, such as the ESS Model 800 (Silvertip). The servers were attached to the ESS via SCSI adapters, but alternatively fibre channel may be used.

### **Installing Advanced Copy Services**

An IBM representative should install Copy Services on the ESS 2105 F20. Copy Services is a separately sold feature of the ESS. It includes FlashCopy for the Open Systems environment.

### **Installing the Command Line Interface on AIX**

Install the ESS Command Line Interface on the AIX server from which you want to execute the FlashCopy tasks. Alternatively, you can also execute the FlashCopy tasks from the ESS Copy Services Web Interface.

Install the Copy Services Command Line Interface software. It requires the IBM 2105 Runtime environment for AIX as a prerequisite.

The installation fileset of the Command Line Interface is named `ibm2105cli.rte`. The destination of the fileset is `/usr/opt/ibm2105cli`. The installation fileset of the IBM 2105 Runtime environment for AIX is `ibm2105.rte`. The destination of the fileset is `/usr/opt/ibm2105`.

### **Creating and assigning ESS logical volumes**

1. Use the StorWatch ESS Specialist Web Interface to define logical volumes and assign them to the Primary and the Backup server.
  - a. Select **Storage Allocation** from the **StorWatch ESS Specialist** menu
  - b. Select **Open Systems Storage -> Modify Volume Assignments -> Add Volumes**
  - c. Select the server, the SCSI adapter to which the logical volume(s) should be attached, and the disk group on which the logical volumes should be created and click **Next**.
  - d. Select **Raid-5** as storage type, specify the size and number of logical volumes and click **Perform Configuration Update**.
  - e. Repeat these steps until you have created and assigned all logical volumes that are required.

Make sure that for each source disk you create a target disk of the same size or bigger in the same ESS LSS.

2. On each AIX server, run `cfgmgr` to define SCSI paths to the ESS. The ESS logical volumes will be visible as `hdisks` on the Primary and Backup servers. Use the following AIX command to list the ESS logical volumes assigned to a server:

```
lsdev -C | grep 2105
```

3. Use the ESS Command Line Interface command `rsList2105s` to display the mapping of host physical volume name to ESS volume serial number for all disks assigned to the host system:

```
cd /usr/opt/ibm2105cli  
./rsList2105s.sh
```

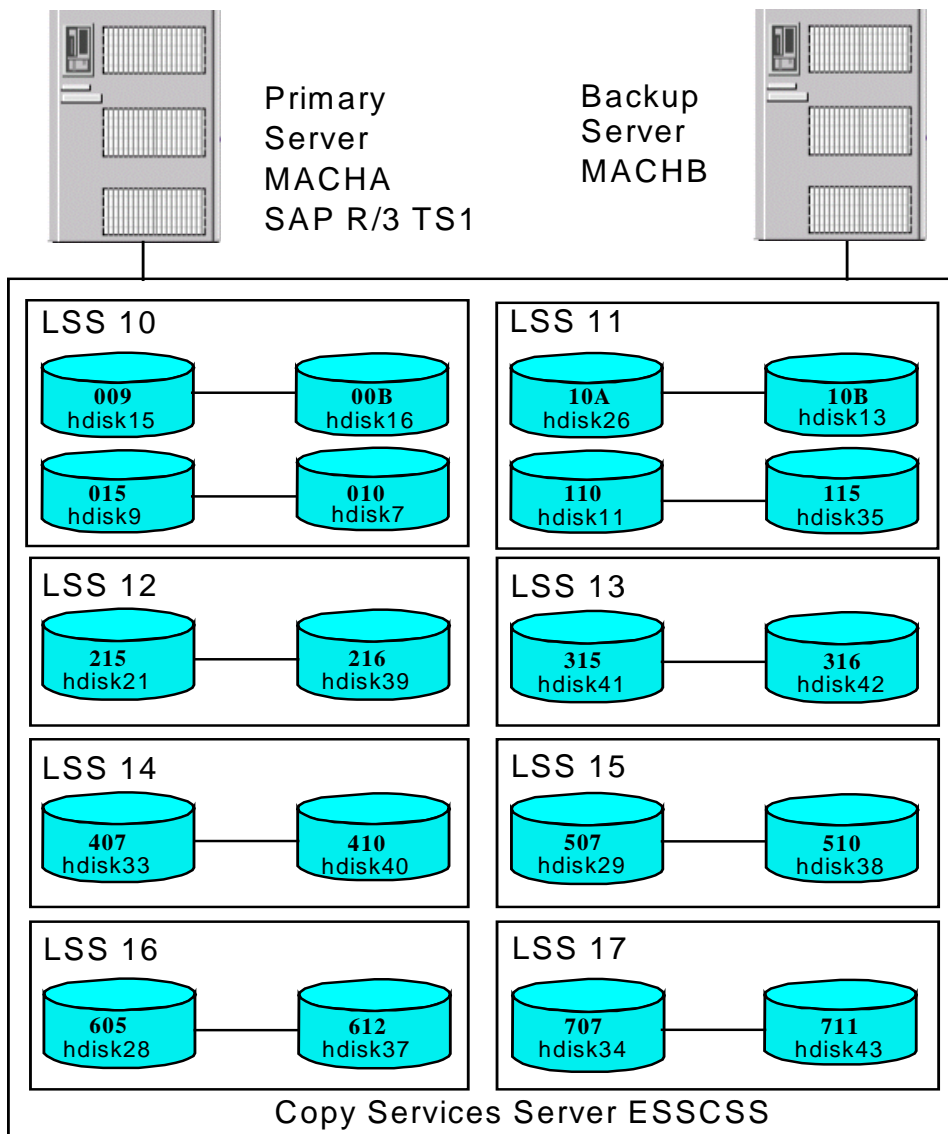
4. Determine the SCSI paths and disks to be used for the SAP system, the DB2 database, and the DB2 logs.
5. Document the corresponding ldev's <<logical devices??>> to be used for AIX logical volumes.

## Configuring a system with two hosts

### Overview

1. Install the ESS CLI on the Primary Server.
2. Assign ESS disks to both servers.
3. Install DB2 for AIX on both servers.
4. Install a standard SAP R/3 system on the Primary Server.

Figure 2 shows the configuration we used for the FlashCopy examples.



**Figure 2. FlashCopy configuration**

The key points to note from Figure 2 are:

- The host name of the Primary server is MACHA.

- The host name of the Backup server is MACHB.
- The host name of the ESS Copy Services Server is ESSCSS.
- The name of the SAP system is TS1; its system number is 00.
- To achieve optimal performance the disks are spread equally over all 8 LSSs of the ESS.
- The Primary server uses the source disks while the Backup server uses the target disks.
- The Primary server accesses the ESS volumes, as follows:

<b>ESS volumes</b>	<b>Logical subsystem</b>
009 and 015	10
10A and 110	11
215	12
315	13
407	14
507	15
605	16
707	17

- 
- These devices are visible to AIX as
  - hdisk15
  - hdisk9
  - hdisk26
  - hdisk11
  - hdisk21
  - hdisk41
  - hdisk33,
  - hdisk29,
  - hdisk28,
  - hdisk34.
- The Backup server accesses the ESS volumes as follows:

<b>ESS volumes</b>	<b>Logical subsystem</b>
00B and 010	10
10B and 115	11
216	12
316	13
410	14
510	15
612	16
711	17

- These devices are visible to AIX as:
  - hdisk16
  - hdisk7
  - hdisk13
  - hdisk35
  - hdisk39

- hdisk42
- hdisk40
- hdisk38
- hdisk37
- hdisk43
- The source and target volume of each FlashCopy pair reside in the same ESS LSS.

**Note:** The root userid must be used to execute all ESS and AIX commands, and to install DB2 and create the DB2 instance. The remainder of the DB2 commands are executed from the ID db2ts1, which is the DB2 instance-owning ID.

1. Create the logical volumes in each LSS and assign them to the Primary Server and the Backup Server via the Web interface of the StorWatch Enterprise Storage Specialist.
2. From each AIX server run `cfgmgr` to define SCSI paths to the ESS.
3. Create a FlashCopy task for the SAP system via the ESS Copy Services Web interface:
  - a) Create the task `ts1sap` to flash the disk on which the SAP software is installed.
  - b) If you defined the task `ts1sap` with the 'no copy' option, you must also define another task called `ts1sapwd` to withdraw the FlashCopy pair for the SAP software. If you did not specify 'no copy' do not create `ts1sapwd`.

We will need a number of FlashCopy tasks for the database file systems to support the different scenarios:

a. Create the tasks `ts1db_10` to `ts1db_17` to flash the logical volumes containing the database data and log files in each LSS. If you create the tasks with the no copy option, you must create appropriate tasks `ts1db_10_wd` to `ts1db_17_wd` to withdraw the FlashCopy pairs.

b. Group the tasks as follows:

Group `ts1db_10` to `ts1db_17` into task, `ts1db`

Group `ts1db_10_wd` to `ts1db_17_wd` into task `ts1dbwd`.

With these tasks FlashCopy pairs for the data and log file systems of the database can be established and withdrawn.

For the Standby Database Scenario we need a FlashCopy task that only handles the database data filesystems without the log filesystem. We group the tasks `ts1db_10` to `ts1db_16` into the group task `ts1dbdata` and the tasks `ts1db_10_wd` to `ts1db_16_wd` into the group task `ts1dbdatawd`. We will also need a FlashCopy task for flashing the log filesystem and for withdrawing the FlashCopy pair.

For the Split Mirror Restore Scenario we need FlashCopy tasks for the database data filesystems only without the 'no copy' option. We also need FlashCopy tasks for establishing reverse FlashCopy pairs for the database data filesystems. Although these could be defined with the no copy option this is not recommended. We create two group tasks: ts1dbdatafc and ts1dbdatarevfc.

The following table lists the FlashCopy tasks that we created:

Task	Description
ts1sap	SAP system establish
ts1sapwd	SAP system withdraw
ts1db	DB data and logs establish
ts1dbwd	DB data and logs withdraw
ts1dbdata	DB data establish
ts1dbdatawd	DB data withdraw
ts1dblog	DB logs establish
ts1dblogwd	DB logs withdraw
ts1dbdatafc	DB data establish (full copy)
ts1dbdatarevfc	DB data reverse establish (full copy)

5. On the Primary server, create volume groups, logical volumes, and file systems to be used by SAP and DB2. Then change the size of the file systems, mount the file systems, and change ownership of the file systems to the SAP administrator user and the DB2 instance owner respectively.

Below is a list of file systems that will be used for storing the data for the examples used throughout this paper:

The file systems for DB2 data are contained in /db2/TS1/sapdata1 to /db2/TS1/sapdata7 -

The file system for DB2's active log files is contained in /db2/TS1/log\_dir

The file system for databases and the database

directory is contained in /db2/TS1/db2ts1/NODE0000 -

The file system for the SAP software is contained in /usr/sap

**Note:** The home directory for the DB2 instance owner must be created manually on each machine and will not utilize any pairs. DB2 archive logs do not have to be placed in a pair. They are treated and stored as you would normally do as part of your recovery strategy and plan.

Here are examples of how to create volume groups, logical volumes, and file systems on the Primary server:

#### **Create volume groups**

```
# smit vg ->Add a Volume Group
```



```

Add a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[VOLUME GROUP name] [sapvg]
Physical partition SIZE in megabytes 64
* PHYSICAL VOLUME names [hdisk15]
Activate volume group AUTOMATICALLY
  at system restart? yes
Volume Group MAJOR NUMBER []
Create VG Concurrent Capable? no
Auto-varyon in Concurrent Mode? no

```

Alternatively, issue the following command:  
 mkgvg -f -y sapvg -s 64 hdisk15

```
# smit vg
->Add a Volume Group
```

```

Add a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[VOLUME GROUP name] [dbvg]
Physical partition SIZE in megabytes 64
* PHYSICAL VOLUME names <disk26 hdisk28 hdisk29 hdisk33]
Activate volume group AUTOMATICALLY
  at system restart? yes
Volume Group MAJOR NUMBER []
Create VG Concurrent Capable? no
Auto-varyon in Concurrent Mode? no

```

Alternatively, issue the following command:  
 mkgvg -f -y dbvg -s 64 hdisk9 hdisk11 hdisk21 hdisk26  
 hdisk28 hdisk29 hdisk33 hdisk34 hdisk41

### Create logical volumes

The following procedure creates two logical group names: saplv and data1lv.

```
#smit lv
->Add a Logical Volume ->VOLUME GROUP name [sapvg]
```

### Add a Logical Volume

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

	[Entry Fields]
Logical volume NAME	[saplv]
* VOLUME GROUP name	sapvg
* Number of LOGICAL PARTITIONS	[60]
PHYSICAL VOLUME names	[]
Logical volume TYPE	[]
POSITION on physical volume	middle
RANGE of physical volumes	minimum
MAXIMUM NUMBER of PHYSICAL VOLUMES to use for allocation	[]
Number of COPIES of each logical partition	1
Mirror Write Consistency?	yes
Allocate each logical partition copy on a SEPARATE physical volume?	yes
RELOCATE the logical volume during reorganization?	yes
Logical volume LABEL	[]
MAXIMUM NUMBER of LOGICAL PARTITIONS	[512]
Enable BAD BLOCK relocation?	yes
SCHEDULING POLICY for reading/writing logical partition copies	parallel
Enable WRITE VERIFY?	no
File containing ALLOCATION MAP	[]
Stripe Size?	[Not Striped]

Alternatively, issue the following command:

```
mklv -y saplv sapvg 60
```

```
#smit lv
```

**->Add a Logical Volume    ->VOLUME GROUP name [dbvg]**

### Add a Logical Volume

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

```

                                     [Entry Fields]
Logical volume NAME                  [data1lv]
* VOLUME GROUP name                  dbvg
* Number of LOGICAL PARTITIONS       [295]
  PHYSICAL VOLUME names              [hdisk9]
  Logical volume TYPE                 []
  POSITION on physical volume          middle
  RANGE of physical volumes           minimum
  MAXIMUM NUMBER of PHYSICAL VOLUMES to use for allocation []
  Number of COPIES of each logical partition 1
  Mirror Write Consistency?          yes
  Allocate each logical partition copy on a SEPARATE physical volume? yes
  RELOCATE the logical volume during reorganization? yes
  Logical volume LABEL                []
  MAXIMUM NUMBER of LOGICAL PARTITIONS [512]
  Enable BAD BLOCK relocation?        yes
  SCHEDULING POLICY for reading/writing logical partition copies parallel
  Enable WRITE VERIFY?                no
  File containing ALLOCATION MAP        []
  Stripe Size?                         [Not Striped]
```

Alternatively, issue the following commands:

```
mklv -y dblv dbvg 60 hdisk26
mklv -y data1lv dbvg 295 hdisk9
mklv -y data2lv dbvg 295 hdisk33
mklv -y data3lv dbvg 295 hdisk11
mklv -y data4lv dbvg 295 hdisk29
mklv -y data5lv dbvg 295 hdisk21
mklv -y data6lv dbvg 295 hdisk41
mklv -y data7lv dbvg 295 hdisk28
mklv -y dbloglv dbvg 295 hdisk34
```

### Create file systems.

```
#smit fs
```

- >Add / Change / Show / Delete File Systems ->Journaled File Systems
- >Add a Journaled File System on a Previously Defined Logical Volume
- >Add a Standard Journaled File System

### Add a Large File Enabled Journaled File System

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

```

[Entry Fields]
* LOGICAL VOLUME name      saplv
* MOUNT POINT              [/usr/sap]
Mount AUTOMATICALLY at system restart? yes
PERMISSIONS                read/write
Mount OPTIONS              []
Start Disk Accounting?     no
Number of bytes per inode  4096
Allocation Group Size (MBytes) 64
```

Alternatively, issue the following command:

```
crfs -v jfs -d saplv -m /usr/sap -A yes -p rw
```

```
#smit fs
```

**->Add / Change / Show / Delete File Systems ->Journaled File Systems -  
>Add a Journaled File System on a Previously Defined Logical Volume  
->Add a Standard Journaled File System**

### Add a Large File Enabled Journaled File System

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.

```

[Entry Fields]
* LOGICAL VOLUME name      data1lv
* MOUNT POINT              [/db2/TS1/sapdata1]
Mount AUTOMATICALLY at system restart? yes
PERMISSIONS                read/write
Mount OPTIONS              []
Start Disk Accounting?     no
Number of bytes per inode  4096
Allocation Group Size (MBytes) 64
```

Alternatively, issue the following commands:

```
crfs -v jfs -d dblv -m /db2/TS1/db2ts1/NODE0000 -A yes -p rw
crfs -v jfs -d data1lv -m /db2/TS1/sapdata1 -A yes -p rw
crfs -v jfs -d data2lv -m /db2/TS1/sapdata2 -A yes -p rw
crfs -v jfs -d data3lv -m /db2/TS1/sapdata3 -A yes -p rw
crfs -v jfs -d data4lv -m /db2/TS1/sapdata4 -A yes -p rw
crfs -v jfs -d data5lv -m /db2/TS1/sapdata5 -A yes -p rw
crfs -v jfs -d data6lv -m /db2/TS1/sapdata6 -A yes -p rw
crfs -v jfs -d data7lv -m /db2/TS1/sapdata7 -A yes -p rw
crfs -v jfs -d dbloglv -m /db2/TS1/log_dir -A yes -p rw
```

Change the size of the file system (it is changed to ~7GB in this example):

```
chfs -a size=14000000 /usr/sap
chfs -a size=14000000 /db2/TS1/sapdata1
chfs -a size=14000000 /db2/TS1/sapdata2
chfs -a size=14000000 /db2/TS1/sapdata3
chfs -a size=14000000 /db2/TS1/sapdata4
chfs -a size=14000000 /db2/TS1/sapdata5
```

```

chfs -a size=14000000 /db2/TS1/sapdata6
chfs -a size=14000000 /db2/TS1/sapdata7
chfs -a size=14000000 /db2/TS1/log_dir
chfs -a size=14000000 /db2/TS1/db2ts1/NODE0000

```

Mount the file systems:

```

mount /usr/sap
mount /db2/TS1/db2ts1/NODE0000
mount /db2/TS1/sapdata1
mount /db2/TS1/sapdata2
mount /db2/TS1/sapdata3
mount /db2/TS1/sapdata4
mount /db2/TS1/sapdata5
mount /db2/TS1/sapdata6
mount /db2/TS1/sapdata7
mount /db2/TS1/log_dir

```

Change ownership of the file systems to the SAP administrative user and DB2 instance owner respectively (users ts1adm in group sapsys and user db2ts1 and group dbts1adm in this example):

```

chown ts1adm:sapsys /usr/sap
chown db2ts1:dbts1adm /db2/TS1/sapdata1
chown db2ts1:dbts1adm /db2/TS1/sapdata2
chown db2ts1:dbts1adm /db2/TS1/sapdata3
chown db2ts1:dbts1adm /db2/TS1/sapdata4
chown db2ts1:dbts1adm /db2/TS1/sapdata5
chown db2ts1:dbts1adm /db2/TS1/sapdata6
chown db2ts1:dbts1adm /db2/TS1/sapdata7
chown db2ts1:dbts1adm /db2/TS1/log_dir

```

6. Install DB2 on both the Primary and Backup server if this is not already done. Further information on installing DB2 can be found in the DB2 UDB documentation.
7. Install the SAP R/3 system as Central Instance (database server and SAP application server on the same system) on the Primary server from the SAP CDs as described in the SAP installation guide with the R3Setup program. Name the SAP system TS1 and use 00 as system number. Before running R3Setup modify the file DBSIZE.TPL in the installation directory as follows: replace
 

```

PSAPTEMP=0;0;32;4;@Z_SAPDATAPATH=SAPDATAT@

```

 with
 

```

PSAPTEMP=1800;1;8;4;@Z_SAPDATAPATH=SAPDATA7@

```

This change will cause the temporary tablespace PSAPTEMP to be created as a DMS tablespace with one container in /db2/TS1/sapdata7. By default, an SMS tablespace is created.

The R3Setup program will create the DB2 instance db2ts1 and the database TS1. It will also update the database configuration to set the log path to

/db2/TS1/log\_dir. Loading the data into the database may last several hours.

For more details about installing an SAP R/3 system please refer to the *SAP R/3 Installation Guide*.

8. Flash the SAP software to the target disk and system:
  - a. Create the group dbts1adm and the user ts1adm on the target system with the same userid and groupid as on the source system.
  - b. Copy the files from the ts1adm home directory (default: /home/ts1adm) to the ts1adm home directory on the target system. Alternatively you can also place the home directory of the ts1adm user on the ESS disk used for the SAP system and create a link.
  - c. Make the following changes to the files in the home directory on the target system:
    - ❑ Change the names of the files .dbenv\_MACHA.csh and .sapenv\_MACHA.csh to .dbenv\_MACHB.csh and .sapenv\_MACHB.csh respectively.
    - ❑ Change the names of the scripts used to start and stop the SAP system: change startsap\_MACHA\_00 and stopsap\_MACHA\_00 to startsap\_MACHB\_00 and stopsap\_MACHB\_00 respectively.
  - d. As root user execute the ts1sap FlashCopy task:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS  
ts1sap
```
  - e. Mount the file system /usr/sap on the target system:

```
mount /usr/sap
```
  - f. Modify the SAP system profiles on the target system in the directory /usr/sap/TS1/SYS/profile as follows:
    - Change the names of the profiles START\_DVEBMGS00\_MACHA and TS1\_DVEBMGS00\_MACHA to START\_DVEBMGS00\_MACHB and TS1\_DVEBMGS00\_MACHB respectively.
    - Edit the profiles DEFAULT.PFL, START\_DVEBMGS00\_MACHB and TS1\_DVEBMGS00\_MACHB and change all occurrences of MACHA to MACHB.
9. Create an identical instance db2ts1 on the Backup server. The HOME file system is not part of a pair and may reside on an internal disk. Ensure the UID/GID of the instance owner is identical across both machines. The current example will use db2ts1 as the instance owner. Since there are two separate copies of the instance directory, ensure that you update database manager configuration parameters and DB2 profile variables on both instances, if required.
10. After installation the SAP system on the Primary server will be started automatically.

11. On the Backup server from the db2inst1 instance, catalog the database TS1 even though the file systems that contain the database TS1 does not currently exist on the Backup server:

```
db2 "catalog db TS1 on /db2/TS1"
```

The base configuration is now done and the following sections will provide examples on how ESS FlashCopy technology and DB2 UDB suspended I/O technology can be used together.

---

## Snapshot database scenario

This scenario starts with SAP R/3 system TS1 running on the Primary server and clients using the Primary database. The Backup server has TS1 and DB2 stopped and DB2 file systems for data and logs unmounted.

1. From the Primary Server issue the suspend command on the Primary database:  

```
db2 "connect to ts1"  
db2 "set write suspend for database"
```
2. Execute the FlashCopy task ts1db for flashing the data and log file systems from the Primary server as root user:  

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS ts1db
```

The `-v` option indicates verbose mode, the `-s` parameter specifies the hostname of the Copy Services Server. Wait for `rsExecuteTask` to return.
3. Issue the resume command from the same connection in step 1:  

```
db2 "set write resume for database"
```
4. From the Backup server, mount the file systems:  

```
mount /db2/TS1/db2ts1/NODE0000  
mount /db2/TS1/sapdata1  
mount /db2/TS1/sapdata2  
mount /db2/TS1/sapdata3  
mount /db2/TS1/sapdata4  
mount /db2/TS1/sapdata5  
mount /db2/TS1/sapdata6  
mount /db2/TS1/sapdata7  
mount /db2/TS1/log_dir
```
5. Start the DB2 instance db2ts1 on the Backup server:  

```
db2start
```
6. The following command performs crash recovery on the database and takes it out of a suspended state :  

```
db2inidb ts1 as snapshot
```

The database on the Backup server is now in a transactionally consistent state at the time the suspend command was issued against the Primary database.
7. Start the SAP system TS1 by logging on as user ts1adm on the Backup server and issuing the following command:  

```
startsap
```
8. To reconfigure the transport system for the new host, log on to the SAP system and execute transaction STMS.

If a new snapshot image is desired, stop the SAP system TS1 on the Backup server and unmount the `/db2/TS1/...` file systems. Then, withdraw the existing FlashCopy pairs by executing the task `ts1dbwd` (if they had been flashed using the `no copy` option). Then re-establish the FlashCopy pairs and execute the above steps again.





---

## Standby database scenario

This scenario starts with the SAP R/3 system TS1 running on the Primary server. On the Backup server, TS1 is stopped and the DB2 filesystems for data and logs are unmounted. This scenario assumes that the FlashCopy pairs and tasks are already defined on the ESS Copy Services Server.

Because the standby scenario requires logs from the Primary database to be applied to the Backup database, configure the SAP DB2 userexit program to copy the log files to the Backup server. Ship the archive logs to the Backup server so that the Backup database can apply these logs. To do this:

1. NFS-mount the log directory of the Backup system (TS1 on MACHB).
2. In the configuration file `initTS1.db6` in the directory `/usr/sap/TS1/SYS/global` set the value of the variable `DB2DB6_STANDBY_DIR` to the NFS-mounted log directory of the target system.

For more details on how to configure the SAP DB2 administration tools, see the SAP manual *SAP Database Administration Guide: IBM DB2 Universal Database for UNIX and Windows* that is also available at <http://service.sap.com/installguides>.

1. Issue the suspend command on the Primary database:

```
db2 "connect to ts1"
db2 "set write suspend for database"
```
2. Flash the data pairs from the Primary server:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS ts1dbdata
```

**Attention:** For this scenario, do not copy or make available to the Backup server the `/db2/TS1/log_dir` file system on the Primary server at this point.
3. Wait for the FlashCopy task to finish.
4. Issue the resume command from the same connection used in step 1:

```
db2 "set write resume for database"
```
5. Mount the data filesystem on the Backup server:

```
mount /db2/TS1/db2ts1/NODE0000
mount /db2/TS1/sapdata1
mount /db2/TS1/sapdata2
mount /db2/TS1/sapdata3
mount /db2/TS1/sapdata4
mount /db2/TS1/sapdata5
mount /db2/TS1/sapdata6
mount /db2/TS1/sapdata7
```

6. Start the DB2 instance db2ts1 on the Backup server:  

```
db2start
```
7. Place the Backup database into a rollforward pending state, and take it out of its suspended state:  

```
db2inidb ts1 as standby
```
8. As DB2 log files are successfully getting archived to the archive directory on the Primary database, make them available to the retrieve path for the Backup database. Be careful not to roll forward through a log file on the Backup server that is only partially copied to the retrieve directory.
9. On the Backup database, roll forward through the DB2 logs that have been archived from the Primary database. The `rollforward` command gives you the options to roll forward to end of logs or to a point in time. Rolling forward to a point in time can be useful if the intention is to keep the Backup database a constant amount of time behind the Primary database:  

```
db2 "rollforward database ts1 to end of logs"
```

or

```
db2 "rollforward database ts1 to 2000-06-19- 19.56.00.0000"
```

**Note:** The retrieved log files are copied to `/db2/TS1/log_dir` by DB2 during the rollforward, which is the current mount point for the log directory. Ensure that the instance owner has write access to this directory and that the directory is large enough. When the log logical volume is copied, and `/db2/TS1/log_dir` is mounted, the contents of the `/db2/TS1/log_dir` mount point will no longer be visible. This does not cause a problem, because any log files that may have been in this directory can still be retrieved from the retrieve path.

10. Continue with iterations of the previous `rollforward` commands until the Primary database fails, a database application logic error occurs, or until use of the Backup database is desired.

If you do not need to roll forward through the most current logs from the Primary system, then proceed to step 16.

11. At this point, ensure all log files that have been archived on the Primary server are available to the Backup server.
12. Copy the log pair from the Primary server:  

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS  
ts1dblog
```

and wait for FlashCopy task to finish.

13. Mount the log filesystem on the Backup server:  

```
mount /db2/TS1/log_dir
```

14. Issue a rollforward stop command on the Backup database. If a database application logic error occurred, the user must rollforward to a point-in-time

before the logic error occurred. If the database needs to be recovered to the point when the Primary server failed, then a rollforward to end of logs should be performed:

```
db2 "rollforward database ts1 to end of logs and stop"
```

or

```
db2 "rollforward database ts1 to 2000-06-19-20.57.00.0000 and stop"
```

15. Start the SAP system: log on as ts1adm user on the Backup server and issue the command

```
startsap
```

---

## Split-Mirror online backup scenario

This scenario starts with the SAP R/3 system TS1 running on the Primary server. The Backup server has TS1 stopped and DB2 file systems for data and logs unmounted.

**Attention:** This scenario is currently not supported for databases that contain SMS tablespaces.

1. NFS-mount the log directory of the system where you want to restore the backup and configure the SAP DB2 userexit program so that it copies the log files to this directory: in the configuration file `initTS1.db6` in the directory `/usr/sap/TS1/SYS/global` set the value of the variable `DB2DB6_STANDBY_DIR` to the NFS-mounted log directory of the target system. For more details on how to configure the SAP DB2 administration tools please refer to the SAP manual *SAP Database Administration Guide: IBM DB2 Universal Database for UNIX and Windows* that is available at <http://service.sap.com/installguides>.

2. Issue the suspend command on the Primary database:

```
db2 "connect to ts1"  
db2 "set write suspend for database"
```

3. Execute the FlashCopy task `ts1db` from the Primary server as root user:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS ts1db
```

Wait for `rsExecuteTask` to return.

**Note:** If you are at DB2 UDB V7 FP4 and above, you do not have to copy the logs for this scenario

4. Issue the resume command from the same connection as used in Step 2:

```
db2 "set write resume for database"
```

5. Mount the data and log filesystems on the Backup server:

```
mount /db2/TS1/db2ts1/NODE0000  
mount /db2/TS1/sapdata1  
mount /db2/TS1/sapdata2  
mount /db2/TS1/sapdata3  
mount /db2/TS1/sapdata4  
mount /db2/TS1/sapdata5  
mount /db2/TS1/sapdata6  
mount /db2/TS1/sapdata7  
mount /db2/TS1/log_dir
```

**Note:** If you are on DB2 UDB V7 FP4 and above, you do not need to mount the `/db2/TS1/log_dir` file system.

6. Start the DB2 instance on the Backup server:

```
db2start
```

7. The following command places the Backup database into a rollforward pending state and take it out of a suspended state:

```
db2inidb ts1 as standby
```

8. Perform a DB2 backup on the Backup server using the supported backup system:

```
db2 "backup database ts1 to /backupfs"
```

9. Stop the DB2 instance on the Backup server:

```
db2stop
```

10. Unmount the /db2/TS1/... file systems on the Backup server:

```
umount /db2/TS1/db2ts1/NODE0000
umount /db2/TS1/sapdata1
umount /db2/TS1/sapdata2
umount /db2/TS1/sapdata3
umount /db2/TS1/sapdata4
umount /db2/TS1/sapdata5
umount /db2/TS1/sapdata6
umount /db2/TS1/sapdata7
umount /db2/TS1/log_dir
```

11. Withdraw the FlashCopy pairs for the database and log files:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS ts1dbwd
and wait for the FlashCopy task to finish.
```

Restore the DB2 backup image taken on the Backup server to any DB2 server, including the Primary server, using the DB2 restore command.

1. Stop the SAP system on the Primary server by logging on as user ts1adm and issuing the following command, which also stops the DB2 instance:

```
stopsap
```

2. Restore the DB2 backup to the desired machine:

```
db2 "restore database ts1 from /backupfs"
```

3. Make sure that logs from the Primary server are available to the machine where the database restore was performed, thus allowing a rollforward through the logs.

4. After the restore is successful, roll forward through the DB2 logs. The `rollforward` command allows the option to roll forward to end of logs or to roll forward to a point in time. Ensure all logs that are required from the Primary server are available to the new system:

```
db2 "rollforward database ts1 to end of logs and stop"
```

or

```
db2 "rollforward database ts1 to 2000-06-19-19.56.00.0000 and stop"
```

5. Start the SAP system TS1 by logging on as ts1adm user and issuing the following command:

```
startsap
```

---

## Split-mirror restore scenario

This scenario starts with the SAP R/3 system TS1 running on the Primary server. The Backup server has TS1 stopped and DB2 file system for data and logs unmounted.

1. Edit the userexit program and place it in the appropriate directory on the Primary server so DB2 knows where archived logs are located. The user exit program will archive the logs.
2. Issue the suspend command on the Primary database:

```
db2 "connect to ts1"  
db2 "set write suspend for database"
```
3. Execute the FlashCopy task ts1dbdata from the Primary server as root user:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS  
ts1dbdatafc
```

Wait for rsExecuteTask to return.
4. Issue the resume command from the same connection in Step 2:

```
db2 "set write resume for database"
```
5. If the Primary database gets corrupted due to a database application error or some other error that prevents the database from being restarted, another FlashCopy task that copies the former target volumes to the former source volumes can then be executed. At this point, DB2 should no longer be running on the Primary server.
6. Unmount the /db2/TS1/... data file systems on the Primary server:

```
umount /db2/TS1/db2ts1/NODE0000  
umount /db2/TS1/sapdata1  
umount /db2/TS1/sapdata2  
umount /db2/TS1/sapdata3  
umount /db2/TS1/sapdata4  
umount /db2/TS1/sapdata5  
umount /db2/TS1/sapdata6  
umount /db2/TS1/sapdata7
```
7. Execute the FlashCopy task ts1dbrev from the Primary server as root user:

```
/usr/opt/ibm2105cli/rsExecuteTask.sh -v -s ESSCSS  
ts1dbdatarevfc
```

Wait for rsExecuteTask to return.
8. Mount the /db2/TS1/... data file systems on the Primary server:

```
mount /db2/TS1/db2ts1/NODE0000  
mount /db2/TS1/sapdata1  
mount /db2/TS1/sapdata2  
mount /db2/TS1/sapdata3  
mount /db2/TS1/sapdata4  
mount /db2/TS1/sapdata5
```



```
mount /db2/TS1/sapdata6
mount /db2/TS1/sapdata7
```

9. Start the DB2 instance on the Primary server:

```
db2start
```

10. On the Primary database issue the db2inidb command to take the database image out of a suspended state and place it into a rollforward pending state:

```
db2inidb ts1 as mirror
```

11. On the Primary database, rollforward through the DB2 logs and stop. The rollforward command allows the option to rollforward to end of logs or to rollforward to a point in time:

```
db2" rollforward database ts1 to end of logs and stop"
```

or

```
db2 "rollforward database ts1 to 2000-06-19-19.56.00.0000
and stop"
```

12. Start the SAP system TS1 on the Primary server by logging on as user ts1adm and issuing the following command:

```
startsap
```

## Summary

DB2 UDB's Suspend I/O technology provides a very powerful mechanism to offload work on to a secondary system, as well as providing a means to plan for disaster recovery. This functionality along with the utilization of the IBM Enterprise Storage Server provides an ideal environment for running an SAP R/3 system.